

QUESTION ONE: Write a complete assembly program that defines an array named CAR consisting of 48 elements of type double word and: Solve each of the 4 parts in a separate loop! [18 pts]

- 1) Randomly generates 48 values in the range -60 to +60 and store them in the array CAR.
- 2) Subtract the 2 words in each element of array CAR and store the result in place of the low order word in the same array element. For example, if a is a double word, then [a]=[a]-[a+2].
- 3) Displays all values of array CAR in Hexadecimal as words separated by a space.
- 4) Moves all elements of array CAR by 20 words up in the memory.

```
INCLUDE Irvine32.inc

        .DATA
        sWord    20 dup(?)
        CAR      sdword 48 dup(?)

        .CODE
MAIN    PROC
; Randomly generates 24 values in the range -60 to +60
        MOV     ESI, OFFSET CAR
        MOV     ECX, LENGTHOF CAR
L1:     MOV     EAX, 121
        CALL    RANDRANGE
        SUB     EAX, 60
        MOV     [ESI], EAX
        ADD     ESI, 4
        LOOP    L1
; Swap the WORDS in each element of array CAR
        MOV     ECX, LENGTHOF CAR
        MOV     ESI, 0
L2:     MOV     AX, WORD PTR [CAR[ESI+2]]
        SUB     WORD PTR [CAR[ESI]], AX
        ADD     ESI, 4
        LOOP    L2
; Display elements of array CAR as words
        MOV     ECX, LENGTHOF CAR
        LEA     ESI, CAR
        MOV     EBX, TYPE CAR / 2
        CALL    DUMPHEX
; Moves all elements of array CAR by 20 words UP in the memory
        MOV     ECX, LENGTHOF CAR
        LEA     ESI, CAR
L3:     MOV     EAX, [ESI]
        MOV     [ESI+40], EAX
        SUB     ESI, 4
        LOOP    L3
        EXIT
main    ENDP
        END     MAIN
```

**QUESTION TWO:** Choose the BEST correct answer for each of the following questions and write its letter symbol down in the table shown below {12 points}

- 1) The flag that is not affected by the instruction: INC ax is  
a) carry b) sign c) zero d) parity e) None
- 2) The statement that produces syntax error during assembly process is:  
a) MOV AX, [EBX] b) MOVX BX, AX c) MOVZX AX, DH  
d) MOVX EXAX, DH e) MOVZX EXAX, DH
- 3) The code: T3 SWORD 20 DUP(?,?), 40 DUP(?);  
MOV AX, LENGTHOF T3; stores the value \_\_\_\_\_ in AX register.  
a) 00A0H b) 0080H c) 0080 d) 60 e) None
- 4) The instruction that stores in AX the third word in X word 200, 3AH, 7, 76F7H is:  
a) MOV AX, X b) MOV AX, X[4] c) MOV AX, X[6]  
d) MOV AX, X+3 e) None
- 5) The instruction that stores 0 in the memory byte pointed by ebx register is:  
a) MOV ebx, 0 b) MOV [ebx], 0 c) SUB [ebx], [ebx]  
d) SUB ebx, ebx e) None
- 6) The statement that produces syntax error during assembly process is:  
a) SUB [SI], [DI] b) ADD AX, BX c) XCHG AX, BX  
d) SUB EXAX, 20H e) INC BYTE PTR [BX]
- 7) The 8-bit value 1010011 represents signed decimal value \_\_\_\_\_ and unsigned decimal value \_\_\_\_\_.  
a) -23, 163 b) 163, -93 c) -35, 163 d) -93, 163 e) None
- 8) In real-address mode, if the physical address is 70000 and the offset is 5AC0, the segment value will be:  
a) 6A54 b) 75AC0 c) 75AC d) 6A540 e) None
- 9) The instruction that swaps the contents of CX register with the word pointed to by ebx is:  
a) XCHG CX, EBX b) XCHG [ebx], CX  
c) MOVX ebx, CX d) SHAP [ebx], CX
- 10) The instruction that requires 2 operands of different sizes is  
a) add b) mov c) movzx d) xchg e) None
- 11) The statement that produces syntax error during assembly process is:  
a) MOV AX, [EBX] b) MOVZX EBX, DH c) INC AX  
d) MOV 20, DH e) MOV DH, 20H
- 12) The type of the DESTINATION operand used in the instruction: MOV [ebx], CX is:  
a) indirect b) Direct c) Indexed d) Immediate e) None

Question #	1	2	3	4	5	6	7	8	9	10	11	12
Answer	A	B	C	B	E	A	D	A	B	C	D	C

**QUESTION THREE:**

(13 points)

(a) Given:

```
CAC sdword 20 dup(?);
NEW sdword 2 * LENGTHOF CAC dup(?)
```

Write **NO more than 6 instructions** to unpack every double word in array CAC into 2 words and store them in array NEW (You have to define array NEW as NEEDED above).

```
MOV ECX, 2 * LENGTHOF CAC
MOV EBX, 0
LA: MOV AX, word ptr CAC[EBX]
MOV NEW[EBX], AX
ADD EBX, 2
LOOP LA
```

(a) Assume UU sdword ?; Give **NO more than 4 instructions** to perform the following:

```
UU = UU - 2*BX - ECX
```

```
MOVSX ESI, BX
SUB UU, ESI
SUB UU, ESI
SUB UU, ECX
```

(b) Given:

```
SAT sdword 80 dup(?) ; any values
FAT sdword 0
```

Write a loop consisting of **NO more than 8 instructions** to store in the dword FAT the sum of all words of SAT.

```
MOV ECX, LENGTHOF SAT
LEA EBX, SAT
CLC
LC: MOVSX EAX, word ptr [EBX]
ADC FAT, EAX
ADD EBX, 2
LOOP LC
```

**QUESTION FOUR:**

[10 pts]

Carefully study the following data definitions and instructions then choose the BEST correct answer for each of the following 10 questions.

```

UT    BYTE    5 dup(50H, 80H), 60H, 11H, 22H, 7FH, 9AH, ?
T1    WORD    6F7FH, 6AC4H, 81CFH, 69CFH, 12A8H, 0
vv     EQU     8-UT
T2     DWORD   69CB3A2CH, 248F7C39H, ?, 725A9033H, 956F14BH
T3     SWORD   5 DUP(1,-1), 10 DUP(3A5CH, 4F7AH)

      MOV      BX, sizeof UT+2
      MOV      AX, WORD PTR T2[6]
      MOV      DX, WORD PTR T1-3
      MOV      CH, SIZEOF T3
      MOV      CL, LENGTHOF T2
    
```

- 1) The instruction that extracts the ELEVENTH byte of T1 and stores the resulting value in ch is
  - a) `mov byte ptr T1+10, ch`
  - b) `mov ch, byte ptr T1+10`
  - c) `movzx ch, byte ptr T1+11`
  - d) `mov ch, T1+11`
  - e) None
- 2) The instruction that unpacks the first word of T1 and stores its high order byte in dh is
  - a) `mov dh, byte ptr T1`
  - b) `movzx dh, byte ptr T1+1`
  - c) `mov dh, byte ptr T1+1`
  - d) `mov dh, T1+1`
  - e) None
- 3) The instruction that extracts the high order byte of the first word of T1 and stores it in ecx is
  - a) `movzx ecx, byte ptr T1+1`
  - b) `mov ecx, byte ptr T1+1`
  - c) `mov ecx, [T1+1]`
  - d) `mov ecx, T1+1`
  - e) None
- 4) The instruction that makes ebx point to the last word of T1 is
  - a) `lea ebx, offset T1+sizeof T1-2`
  - b) `lea ebx, T1+sizeof T1-2`
  - c) `lea ebx, T1+lengthof T1-2`
  - d) `lea ebx, T1`
  - e) None
- 5) The instruction that packs the first 2 words of T3 and stores the resulting value in ecx is
  - a) `movzx ecx, dword ptr T3`
  - b) `movsx ecx, word ptr T3`
  - c) `mov ecx, T3`
  - d) `mov ecx, dword ptr T3`
  - e) None

After executing the above instructions

- 6) The value assigned to the constant name vv is:
  - a) 16
  - b) 40
  - c) 1CH
  - d) 28H
  - e) None
- 7) The register cx will contain:
  - a) 3C14H
  - b) 143CH
  - c) 6020H
  - d) 2060H
  - e) None
- 8) The register ax will contain:
  - a) 248FH
  - b) 9033H
  - c) ????
  - d) 8F24H
  - e) None
- 9) The register bx will contain:
  - a) 0016HH
  - b) 0020
  - c) 0022
  - d) 0020H
  - e) None
- 10) The register dx will contain:
  - a) 7F22H
  - b) 2211H
  - c) 9A7FH
  - d) 7F9AH
  - e) None